AN EMBEDDED TRAINING SOLUTION: FBCB2/TACTICAL DECISION MAKING INTELLIGENT TUTORING SYSTEM

Richard H. Stottler
Stottler Henke Associates, Inc.
San Mateo, CA 94402

Bill Pike
US Army STRICOM
Orlando, FL 32826-3276

Abstract

We are developing for STRICOM an Intelligent Tutoring System (ITS) for tank and mechanized infantry company commanders that teaches tactical decision making and the tactical use of FBCB2, a C4I system. These are complex cognitive tasks that normally require the availability of an instructor. This prevents the effective use of embedded systems for training in the field, where an instructor is not typically present. Our ITS interfaces to a tactical simulation and FBCB2 and assumes the duties normally performed by the instructor.

Instructors and experts both agree that company commanders need to improve their tactical decision-making and that this requires more tactical decision-making practice. Practice should include a mix of tactical planning and tactical execution in dynamic simulations that provide 3-D virtual terrain views. Additionally, FBCB2 training decays very quickly so that an embedded, scenario-based training aid would substantially increase combat readiness.

The ITS addresses these problems by teaching tactical decision-making and the proper tactical use of FBCB2 by presenting course material and examples, then testing the commander in tactical situations simulated by OneSAF Test Bed (OTB) and displayed in FBCB2. The ITS first evaluates the student's plan, entered as an FBCB2 overlay and provides an automatic critique. It then monitors the student's actions in the simulated scenario, assesses their correctness for the current situation, and debriefs the student by automatically assembling an After Action Review (AAR). It then infers the knowledge deficiencies of the student, and formulates a remedial instruction plan, which normally includes further course material, examples, and further exercises to practice and test the student's weaknesses.

This paper will first describe the requirements for an embedded training system, give the general capabilities of ITSs and explain why ITSs meet the embedded training requirements, describe the FBCB2/Tactical Decision Making ITS, list the lessons learned from this effort and conclude with work planned for the future. The ITS description begins with an overview, followed by a description of the ITS's functionality, followed by a description of each component, and ends with a description of how evaluation is automatically performed and modeled.

Bibliographic Sketches:

Dick Stottler co-founded Stottler Henke Associates, Inc. (SHAI), an artificial intelligence consulting firm in San Mateo, California in 1988. He has been principal investigator on a number of tactical decision-making intelligent tutoring system projects conducted by SHAI and is working on an ITS for battalion commanders at the Command General Staff College and a prototype for the future combat system, both for the US Army STRICOM. He has a master's degree in computer science from Stanford University.

Bill Pike is the Lead Principal Investigator for Advanced Distributed Learning (ADL) with the US Army STRICOM. At STRICOM, he has been the principal investigator on several ADL topics on integrating courseware with advanced topics, including the use of game engines as assessment tools for tactical principles. He is an officer in the Naval Reserves and has a Master's degree from the UCF in Computer Engineering.

Report Documentation Page

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Embedded Training Solution: FBCB2/Tactical Decision Making Intelligent Tutoring System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Simulation, Training & Instrumentation Command,Orlando,FL,32826-3276**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **11**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

AN EMBEDDED TRAINING SOLUTION: FBCB2/TACTICAL DECISION MAKING INTELLIGENT TUTORING SYSTEM

Richard H. Stottler
Stottler Henke Associates, Inc.
San Mateo, California

Bill Pike
STRICOM
Orlando, Florida

EMBEDDED TRAINING REQUIREMENTS

There were two problems that we were tasked with solving. The first was that C4I System training decays very rapidly. According to Brigadier General Lynch, the commander of the Army's first digitized brigade, after two months away from Force XXI Battle Command Brigade and Below (FBCB2) his soldiers could make no effective use of the system. There are several reasons for this. First, this is a common phenomenon of many software systems. Second, FBCB2 is a very complex system designed for a diverse set of users. Most types of users will use only a small fraction of the system's capabilities. Third, to maintain readiness requires that soldiers have the ability to practice their use of the system in realistic scenarios. The instructors at Ft Hood, where FBCB2 use is trained, also were strong proponents of a scenario-oriented training capability.

The second problem we were tasked with solving is that there needs to be an improvement in the tactical decision-making abilities of company commanders. Discussions with instructors at the Armor School at Fort Knox and observations of students in simulated scenarios showed that the students make many more tactical mistakes and of a more basic nature than expected. Also, there was a high degree of tactical skill variability among the instructors themselves.

Tactical decision-making for company commanders falls naturally into three stages. In the first, the commander receives orders, which include a description of the mission, its objectives, commander's intent and the friendly and enemy situation to the degree that it is known. The student must understand the information and formulate a concept of operations, taking into account the actual situation, which may be different from that assumed by the orders, and the commander's intent. Based on the student's concept of operations, the second stage begins: detailed planning. After the plan is complete, disseminated, and, if possible, rehearsed, the student must execute it. Execution comprises the third stage.

Tactical experts and instructors agree that, for improving tactical decision-making and the tactical use of C4I systems, there is nothing more important that getting practice in scenarios. Practice is required for all three stages. To allow such practice for operational company commanders requires an embedded training system which includes both planning and simulation components. It also requires some type of evaluation and feedback mechanism since meaningful practice requires feedback normally provided by an instructor. But an instructor will not normally be present where the embedded training system will normally be used (in the field) so there exists a need, for all embedded training systems, for an automatic evaluation and feedback system.

WHY AN ITS

Scenario-based Intelligent Tutoring Systems are specifically designed to evaluate decisions made in a scenario and provide feedback to the student. ITSs are artificially intelligent software systems that seek to replace or augment an instructor and provide to the student a tailored, one-on-one tutoring experience. As such, they perform many of the operations normally performed by an instructor. These include monitoring student performance in simulated scenarios; evaluating decisions; providing feedback; inferring strengths, weaknesses, skills and knowledge mastery; and formulating an instructional (or remedial) plan based on these.

Thus ITS technology was considered appropriate for meeting the embedded training requirements. The ITS can tutor on the aspects, and just those aspects, of FBCB2 that company commanders normally use. These include creating FBCB2 plan overlays before the mission and monitoring the tactical situation using

FBCB2 during mission execution. The ITS can also tutor the process of determining the concept of operations, especially monitoring the application of general tactical principle to the concept of operations. It would tutor the development of the detailed plan. Finally, it can tutor the real-time decision-making needed for tactical execution. Tutoring, for each stage, entails informing the student of the relevant knowledge, if not already known, providing the student with appropriate scenarios to test the ability to apply that knowledge, evaluating performance in those scenarios, and remediating problems uncovered.

FBCB2/TACTICAL ITS PROTOTYPE DESCRIPTION

Overview

The FBCB2/Tactical Decision Making ITS for Company Commanders is divided into three sections – FBCB2 use, Planning, and Mission Execution. Each section is a prerequisite for the next. The first section tutors on the use of FBCB2, primarily the creation of plan overviews. It is a prerequisite for the second section, planning, since the student is required to enter plans in the planning section using FBCB2. The planning section tutors the application of general and specific tactical principles to the development of the concept of operations and the detailed plan. The student enters a plan using FBCB2; the ITS provides feedback on the overall concept of operations that the student has chosen as well as the plan details. It is a prerequisite for the third section since the student should be able to apply tactical principles in the relatively loosely time-constrained planning stage before being forced to apply them in real-time decision-making. The third section, execution, tutors the student on tactical mission execution. The student performs the mission in a real-time tactical simulation and the ITS evaluates the decisions made and gives the student a debriefing with a description of which were correct, which were incorrect, and why.

The overall architecture is shown below. The student interacts separately with the OneSAF Test Bed (OTB) simulation, FBCB2, and the ITS. The ITS manages the instructional process, maintains a model of the student's ability and knowledge, provides needed information to the student, extracts information from the student when needed, and evaluates the results of the use of FBCB2 and the simulation. FBCB2 provides two main services onboard various vehicles in a tactical environment. Most importantly it helps the commander to maintain situational awareness by displaying friendly and spotted enemy vehicles on its 2-D dynamic tactical map. It also allows the creation, editing, and sending of electronic messages from vehicle to vehicle, including plan overlays. The simulation that we used for this work was OTB (OneSAF Test Bed). OTB was derived from ModSAF. OTB provides a user interface (UI) for controlling friendly vehicles in the simulation and models the vehicles, terrain, and weapons effects on the vehicles. There are several third party 3-D viewers available for OTB and one was chosen and interfaced to OTB.
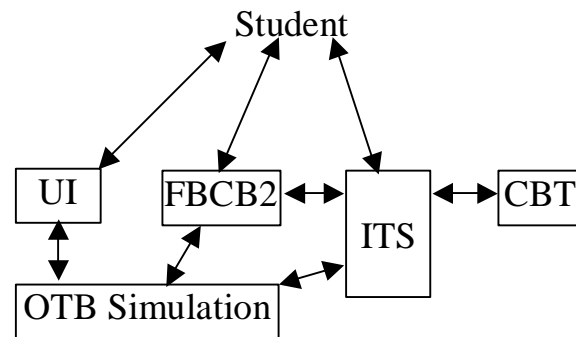

Figure 1. FBCB2 Tactical Decision-Making ITS Architecture

There are several interactions between the modules in the architecture, as shown above. When the ITS determines that the next appropriate instructional event is the presentation of multi media descriptions, such as traditional computer-based training (CBT) files, these are presented to the student. The ITS will select, for some students, appropriate FBCB2 overlay scenarios, brief the student and send these to FBCB2. After receiving the briefing from the ITS, the student enters the overlay into FBCB2. The ITS then reads and evaluates this overlay and provides a multi-media debriefing which might also reference additional CBT files. Planning scenarios are performed in an identical fashion, using FBCB2 to enter plans. Finally the ITS will select appropriate execution scenarios for the student. The student is given a briefing, which may include hints, by the ITS, then the scenario is sent to the simulation and the student performs the mission in the simulation, using the simulation's user interface (UI in Figure 1). The simulation is interfaced to FBCB2, so that the commander can use that system to maintain situational awareness during simulated scenarios. While the simulation is running, the ITS creates a log file of significant events. After the scenario completes, the ITS examines the log file, evaluating each observable decision. It then provides the student with an after action review (AAR).

ITS Functional Description

When a new student logs on, questions are asked about the student's background, experience, and FBCB2 training/use. These questions include level of education achieved, rank, highest unit commanded, types of units served in, familiarity and comfort with computers, familiarity and comfort with FBCB2, and general perceptions as to its usefulness. The ITS uses this information to make initial estimates as to the student's mastery of various principles (including both tactical knowledge and the use of FBCB2) and to select the best instructional method for the student. It is also used to select scenarios, other exercises, types of hints, and other forms of instruction. Mastery categories are Beginner, Novice, Intermediate, and Expert. The Beginner category for a principle occurs when a student performs successfully with it less than 20% of the time. (Novice – 20 to 50%, Intermediate – 50 to 75%, Expert > 75%.) Students at the Intermediate or Expert level for a principle are never given hints. These mastery estimates are available for viewing at all times to the student.

If the ITS estimates that the student's mastery of FBCB2 principles is low, the student is put through FBCB2-only refresher exercises. An introductory lesson explains with detailed steps how to create an overlay and find and place the most relevant symbols.

After the FBCB2 refresher exercises, the ITS begins tutoring the student on general tactical principles. If it estimates that the student's mastery is relatively high it proceeds immediately to tactical decision games (TDGs) presented and answered as FBCB2 overlays. If not, it will first present General Tactical Principle Courseware. For each TDG, the ITS analyzes the student's plan (given as an FBCB2 overlay) and automatically creates a debriefing describing what parts of the plan are right, what parts are wrong, and giving an expert's rationale for the best options. For poor decisions, the ITS will lower its estimate of the mastery of principles related to those decisions, and provide remedial materials on those principles before presenting any more TDGs. The student's overlay is evaluated by comparing it to overlays input by an instructor for that particular TDG. These typically represent a few possible right answers and a few common mistakes. The instructor also has annotated the overlays with information for use by the ITS in assembling the debrief and determining in which principles the student is weak.

For the TDGs and the dynamic execution scenarios, the ITS initially selects exercises based on the need to test untested principles, following each by a debriefing and detailed information on the principles missed. The ITS then begins to retrieve scenarios that exercise the principles in which the student's mastery is weakest. Furthermore, for any scenario using principles that the ITS believes the student is weak in, hints are provided for the scenario. These are generally questions designed to get the student to think about the most important tactical principles required in the scenario.

After the student has demonstrated mastery of general tactical principles in the TDGs, the next part of the course requires that the student know how to apply these and other principles in a dynamic tactical simulation (OTB with a 3-D virtual reality viewing capability). Additionally, more operations-oriented principles (e.g., knowing when and how to use a bounding overwatch) are also tested. The ITS gives the student a short situation description. The student then proceeds to execute the mission in OTB. After the scenario ends, the event log is analyzed by the ITS to determine which actions were correct, incorrect, or omitted, and the underlying principles that were understood and applied or missed. Some scenarios in particular will test the student's use of FBCB2 to maintain situational awareness.

In the dynamic execution scenarios, unplanned actions may occur, such as unexpected contact with the enemy. The student's tanks and other vehicles will begin to react and the student will issue particular orders. The correctness of the student's decisions can be evaluated from the movements and actions of the company's vehicles. As the company commander is also the commander of an individual vehicle, the student is also evaluated on the maneuvers of the company command tank (e.g., is the company commander vehicle being overly endangered, is it with the main effort, and whether the student commander is at the center of a wedge formation).

After the scenario, the commander is debriefed with an After Action Review. All actions are reviewed for correctness and the student is informed about the relevant principles. For the failed principles the student is given detailed information and possibly an example for each. The mastery level estimates for all principles involved are updated. Based on these, a new scenario is retrieved. Scenarios will be selected that test untested principles and test recently failed principles. This process continues until the student's performance in scenarios shows mastery of all required principles.

The prototype has different instructional methods for students with little mastery or experience compared to students with significant mastery and experience. Those with less mastery are given more information prior to running scenarios, more information for each

scenario, and have less choice about what scenario or other instruction they can run next. Those with more mastery are given less information, harder scenarios, and more freedom to choose.

FBCB2 and OTB Description

As described in the architecture, the system contains three major components – FBCB2, OTB, and the ITS itself. An FBCB2 screen capture is shown in Figure 2. FBCB2's major contribution to tactical performance is its display of friendly and enemy spotted vehicles. Each vehicle's individual FBCB2 system must be setup correctly for this to occur. The proper setup causes the vehicle to automatically report its sensed GPS position. Generally, scout vehicles use FBCB2 along with laser range finders to input spotted enemy vehicle positions.
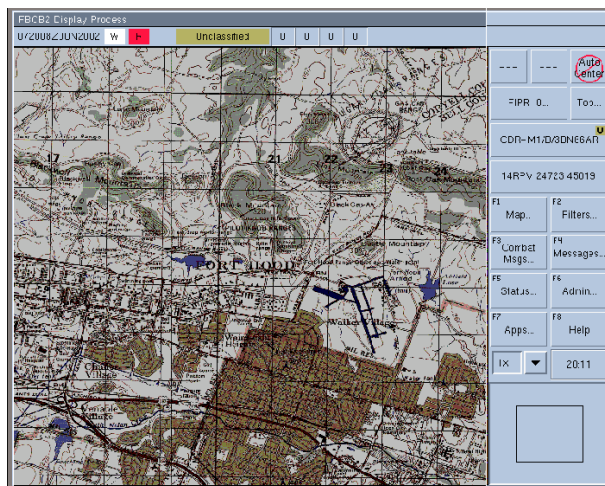

Figure 2. FBCB2 Screen Capture

Company commanders report that during engagements, they still spend 90 to 95% of their time viewing the actual scene and only 5 to 10% updating their situational awareness with FBCB2. FBCB2's combat messages are not generally used during combat by company personnel. The messaging capability provided by FBCB2 is very useful before the mission, however, to distribute orders and plans. While FBCB2 is windows-based and each feature is generally user-friendly when considered in isolation, the sheer volume of its capabilities makes use of individual features related to messaging difficult. Plan overlays are just one type of dozens of possible message types. Knowing how to start editing a plan overlay is difficult, especially if the user is unfamiliar or a significant period of time has passed since the student's last use. A company commander would usually want to use only a few dozen different symbol types, yet FBCB2 contains thousands of them, so knowing where to find a required symbol is difficult. The current version of FBCB2 has many symbols that do not possess the required

flexibility, such as the ability to orient a support by fire position in different directions or to change the width of a ground axis of attack symbol. Additionally, FBCB2 runs under Solaris on a Pentium PC. Since the other components currently run under Windows (or Linux), this required a minimum of two different machines to run the entire system. Finally, FBCB2 is difficult to use in a training context since it is designed and built to run on-board a vehicle, with a GPS and a radio network.

An OTB screen capture is shown in Figure 3. It accurately models the terrain and fire effects upon and the capabilities of a diverse set of vehicles and provides an interface for controlling the friendly vehicles under the student's command. Unfortunately the interface is complex. It often requires training on its usage, resulting in operator training for a simulation required for student training. For a company commander student, who would normally be giving orders to platoon leaders, intelligent platoon level behaviors are lacking. A DIS/HLA gateway was required to provide the HLA data stream that was necessary for the HLA interface to the ITS. The terrain databases available for OTB are somewhat limited. All the 3-D viewers used in concert with OTB require separate terrain databases to meet their 3-D display needs. Thus, to use a piece of terrain for training requires that that terrain be represented in both the OTB terrain database and in the 3-D viewer's database. The amount of terrain in this intersection is severely limited for all viewers we examined. Finally, OTB's modeling of how well a simulated vehicle can see through vegetation often was incompatible with the modeling provided by the 3-D viewer so that many times the simulated vehicle in the simulated world could see another vehicle through vegetation (and therefore fire on it) but the human student using the viewer could not and vice versa.
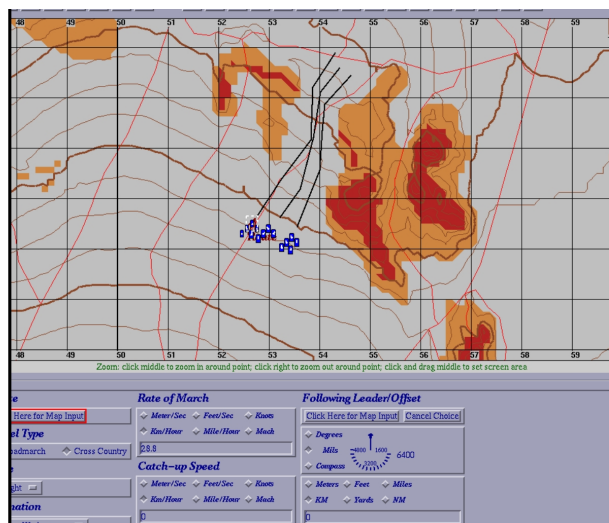

Figure3. OTB Screen Capture

ITS User Interface

The ITS user interface includes 5 panes for:

1) Course Explanation
2) Student Tasks
3) Course Map
4) Student Progress
5) Student History

The Course Explanation provides a brief description of the course. More detailed explanations are also available to the student.

The Student Tasks box holds the current tasks that are available for the user's selection. The format is:
<Action>: <Chapter/Section/Principle name>

There is no enforced order; all the items in the Tasks box are selectable. Actions can be executed either from Student Tasks or Course Map. The ITS provides only actions that correspond to instructional events that it believes are appropriate for this student at this time.

The Course Map, Student Progress, and Student History panels can all expand to fill the right hand column, as shown in Figure 4 for the course map.
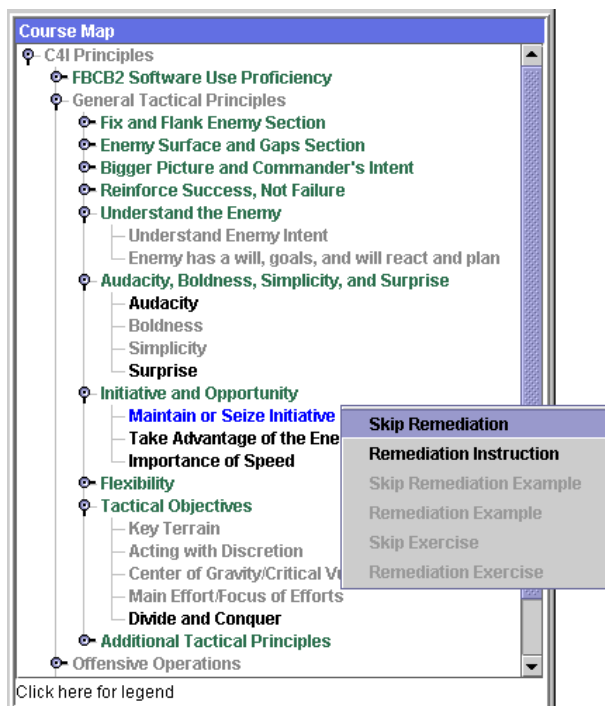

Figure 4. ITS Course Map Pane

The Course Map box, shown in Figure 4, provides the student with a snap shot of the course. It is color coded in the following way:

- **Green** -- Chapter/Section passed

- Gray -- Chapter/Section/Principle is not selectable
- **Black** -- Chapter/Section/Principle is selectable
- **Blue** -- Chapter/Section/Principle is selected

In Figure 4, the student has passed several principles (e.g. FBCB2 Software Use Proficiency, Fix and Flank Enemy Selection, etc.) and can now select one of five principles (e.g. Maintain or Seize Initiative). When the student selects a principle, a list of available actions involving that principle will appear.
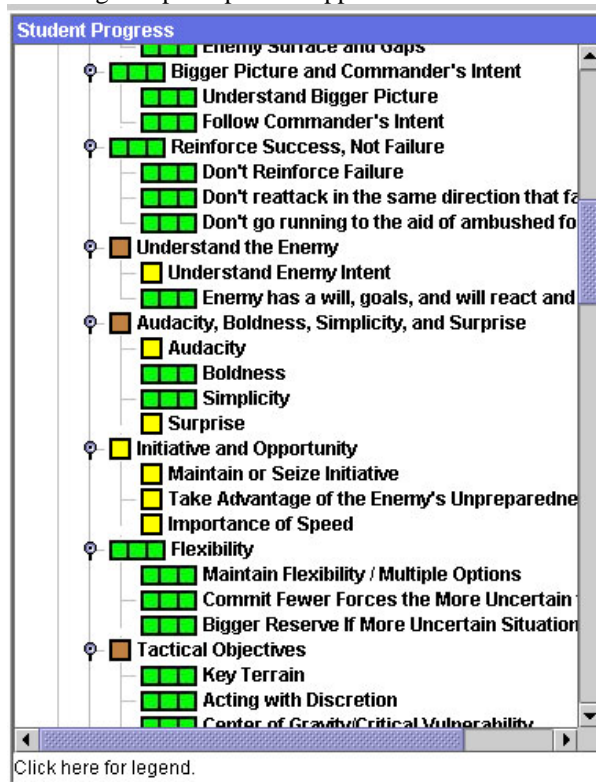

Figure 5. ITS Student Progress Pane

The Student Progress Pane, shown in Figure 5, provides the student with a chart of his progress in the course. It is color coded in the following way:

- ▇ -- Beginner

- ▢ -- Novice

- ▇▇ -- Intermediate

- ▇▇▇ -- Master

- ▢ -- Mixed progress levels exist in this section/chapter

In the above diagram, the student is, in the majority of sections, a Master. In the "Audacity, Boldness,

Simplicity, and Surprise" section, there is mixed expertise, since the student is a Novice at "Audacity" and "Surprise" but a Master in "Boldness" and "Simplicity."

The Student History displays the student's past tasks in reverse chronological order.

FBCB2 Plan Creation Evaluation

In order to verify that students know how to use the FBCB2 plan overlay editor, the ITS requests that they use FBCB2 to create overlays with the symbols they need to use placed at various locations, described in absolute terms or in reference to terrain features or other symbols. The ITS then compares the student's overlay to the correct solution. If an overlay file is successfully completed, the student is credited with knowing how to perform that action. If the student correctly places each symbol in the correct location, credit is also given for knowing where to find that symbol and knowing how to place symbols in the overlay. Failure of any of these principles will prompt the system to debrief the student with detailed instructions on how to perform the type of action that was missed.

Tactical Plan Evaluation

As described above the student will be tasked with solving a TDG with an FBCB2 Plan overlay. The ITS will critique both the plan concept and its details. The ITS uses a Case-Based representation for TDG knowledge. Associated with each TDG scenario is a series of plans previously input by the instructor which represent correct and common incorrect plan solutions. The ITS finds the stored plan that most closely matches the student's entered plan by calculating a plan similarity using a symbol by symbol similarity comparison that considers the type of symbol, the size of unit associated with the symbol, and its general and precise location in reference to terrain features. The instructors have previously annotated each plan with the optimality of the plan, the rationale for the level of correctness of the plan's general concept of operations, and also, which principles should be passed or failed if this plan closely matches the student's plan and detailed descriptions for each symbol in the plan. For each symbol the instructor has entered whether this symbol is correct or a mistake, the priority and importance of this symbol to the overall plan, and various rationale. In the case of the symbol being correct, the rationale includes why this symbol is needed in a good plan, why the type (e.g., armor, infantry) was selected, why the general location was chosen, and why the specific location was chosen. If this symbol represents a

common mistake, the same types of rationale as to why it is a mistake have been entered. The instructor also lists the principles that should be passed or failed if a student's symbol is similar to the particular symbol in the stored plan. The descriptions and rationale listed above are all in the form of multi media files, so the explanations and rationale can include the appropriate graphics and animations as well as text.

The ITS is thus able to assemble a debriefing by first finding the most closely matching plan. If none is found (which is rare, since students usually make the same concept of operations mistakes) then the debriefing merely explains to the student the most optimal plan both in terms of the overall concept and the detailed rationale for each symbol. If the student matches a mistaken plan, a detailed debriefing will appear as to why the assumed concept is poor and which individual symbols were good and bad and why. This is one of the more common occurrences. If the student's plan matches one of the correct plans, then each individual symbol is compared to the corresponding one in the correct plan in terms of type, size, and general and exact location. For any of the symbols not matching the expert's plan on one or more of these dimensions, the student is given the expert's rationale for the correct choice.

Tactical Execution Evaluation

The evaluation of decisions in a free play tactical simulation is one of the most challenging aspects for an ITS. This is because the same scenario can play itself out very differently for different students since students can take any action at any time. Therefore, the student's actions cannot simply be compared to a script of expected actions, since the situation at any time may be very different than that expected by the script.

We've found that in several different domains a generalization of Finite State Machines (FSMs) is very useful for evaluating decisions made in real-time, free-play, simulated, tactical scenarios. These FSMs may be general or scenario specific. They can be executed in real-time, in parallel with the simulation, if a real-time interface exists (using HLA for example) or they can be executed afterward on a log file that includes all the necessary events and vehicle motions.

Typically the evaluation is performed by executing a large number of FSMs simultaneously, where each examines the actions from the perspective of one or a few principles. If available in the interface or log, the student's actions and orders can be evaluated directly. Otherwise, evaluating the events and movements of vehicles usually suffices.
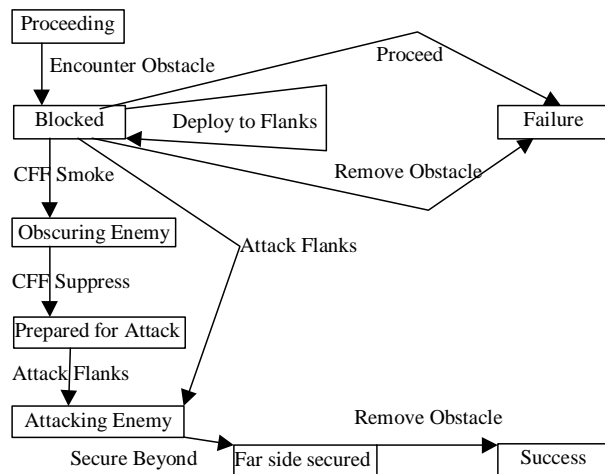
Figure 6. Example Evaluation FSM

An example of an Evaluation FSM is shown in Figure 6. The rectangles represent states and the arrows represent transitions. A transition indicates that the FSM should transition to the next state when the transition's event occurs and its conditions are met. The events and conditions associated with a transition may be quite complex but their meaning is usually summarized by the transition's label. An Evaluation FSM's transitions also include which principles should be passed or failed when the transition is followed and a parameterized message to be written into the debriefing log.

The example in Figure 6 is a simplified FSM which evaluates the proper response to encountering a manmade obstacle while proceeding down a road or narrow avenue. This FSM is one of dozens that would be simultaneously active

The first state is simply labeled "Proceeding" and the FSM starts out in this initial state. When a manmade obstacle is encountered this FSM transitions to the "Blocked" state. Based on the student's actions, a number of different things may occur. The student might simply proceed forward, in which case the FSM will follow the "Proceed" transition. This transition includes the principles that should be failed if the link is followed such as "Understand enemy intent" (i.e., why would the enemy place an obstacle without also placing forces to cover it?) and knowing how to react to manmade obstacles. The transition would also cause a message to be written into the debriefing log indicating that simply proceeding was a mistake and why. Similar principles would be failed if the student immediately tried to remove the obstacle. The student would get some credit if the decision was made to immediately attack likely enemy locations overwatching the obstacle. The credit would be given for "Understand

enemy intent" but the "Obscure an attack" and "Suppress enemy during an attack" principles would have been failed since the student attacked without first calling for obscuring and suppressive fire.

If instead, the student deploys properly to the flanks, the looping transition would have been followed and credit would be given for "Modeling a thinking enemy" and "Defending properly against a likely flank attack". Furthermore, if the student correctly calls for obscuring and suppressive fire, those principles in this scenario would be passed. If the actions of the student cause the FSM to follow all the transitions to the "Success" state, then the last transition ("Remove Obstacle") would include the knowing how to react to manmade obstacles principle as one that was passed.

Dozens of these FSMs execute in parallel, evaluating the student's actions from different perspectives, writing messages to a debriefing log, and sending the ITS lists of failed and passed principles.

Student Modeling

As described above, among other things, each evaluation module compiles lists of passed and failed principles associated with the student's actions and decisions. After each scenario or other instructional event (such as viewing descriptive files) the ITS, for each principle, examines the student's entire history in reference to that principle and determines the level of mastery of the student with respect to that principle. This collection of mastery estimates for all skills and principles is called the student model, since it models the student knowledge and skills. The updates to the student model are immediately reflected in the Student Progress pane of the ITS's user interface. This student model, derived primarily from the student's performance in scenarios, drives the instructional decisions of the ITS such as instructional method, remediation, and example and exercise selection.

Remediation

The ITS provides several forms of remediation in response to deficiencies it observes. When debriefing a student's performance in a scenario, scenario-specific explanations of the student's mistake are given, which reference and apply the general principles to the specifics of the scenario. For example, if a student decides to rush a unit to the direct aid of an ambushed unit he will be failing to honor the general principle, "Don't reinforce failure," but the student will be told how the specific decision made in this scenario violated this principle.

Additionally, if a student consistently has problems applying a principle, summary information describing that failed principle will be given. Further problems by the student will prompt the ITS to provide a more detailed description as well as examples that illustrate the application of the principle in various scenarios.

Students having problems with a principle may also receive hints relating to that principle before beginning a scenario. These typically take the form of a question such as, "What would you do if you were the enemy in this situation?"

The primary limitation of this form of remediation is that it consists of the dynamic assembly of static explanations, rationale, and descriptions that are both scenario specific and general. In other words, although the system dynamically determines what content to display and can assemble different small components targeted to very specific problems, those components must already exist.

System Interfaces

Due to contract and demonstration requirements, there were a very large number of different systems that needed to be interfaced. This caused a significant number of problems. Only the major software interfaces will be described here. Probably the weakest link was the FBCB2 interface to OTB that allowed the real-time simulated scenario from OTB to be viewed in FBCB2. An existing product, SATIDS, which was originally designed to model tactical networks, was used since it already could read and produce both DIS (used by OTB) and VMF (used by FBCB2) messages. Unfortunately, since SATIDS was not designed to be a simple interface between these message formats, it is difficult to setup and brittle. For example, minor changes to STRICOM's network would render the interface inoperable.

The next most problematic interfacing software related to the license manager used by Mak Technology. This was important because we used both Mak's DIS/HLA gateway as the part of the interface between OTB and the ITS and Mak's 3-D viewer to allow students to see a dynamic 3-D display of the terrain and vehicles. Periodic problems with the license manager often prevented both of these from working. When the license manager was working, the gateway caused few problems and the code to create the HLA log and then convert it to a form useable by the ITS was straight-forward to implement. We found that it was a good idea from a processor load perspective to run OTB and the 3-D viewer on separate machines.

The software to convert FBCB2 overlays into a format usable by the ITS was straightforward, once we had obtained the proper library to decode VMF messages. Obtaining the proper library took a surprising amount of time.

STRICOM had originally intended to run a Linux-based version of OTB; however, the PC that created the OTB scenarios was Windows-based. These scenarios could not be executed on a Linux-based OTB PC, thus forcing STRICOM to use Windows to host OTB.

Lessons Learned

We learned a number of lessons from this effort. They can most easily be divided into three categories: those relating to training needs, those relating to automated training techniques, and those relating to the systems that we were working with.

It is clear that more tactical decision-making practice is needed. Tactical proficiency should be improved considerably among tactical practitioners, students and instructors. Since software skills decay quickly there is a need for both initial and refresher scenario-based training to allow far more C4I system tactical scenario training. For both of these, experts and instructors agree that there is nothing more important that getting practice in scenarios. Furthermore, this practice must be accompanied by an expert debriefing. When scenarios are merely played against others at a similarly low level of expertise, students learn very little, since their mistakes are not readily apparent to them. In general debriefing needs to be improved and made available automatically so that students can practice away from the schoolhouse.

Because of the universal agreement for more practice and feedback, experts and instructors are very accepting of the ITS concept with a couple of constraints. The simulation to play the scenarios must be very easy to use. Neither student nor instructor wants to spend time learning how to use a simulation system that is only used in training. Secondly and similarly, the ITS must be user-friendly and intuitive, requiring little or no time to learn.

There tends to be a very high degree of similarity between students' right and wrong answers in tactical scenarios. This makes development of an automatic evaluation system far more straight forward, especially with the cooperation of the instructors in developing scenarios for automated evaluation.

Tactical decision-making falls naturally into three phases - concept of operations, detailed planning, and execution. It is most helpful to evaluate and debrief each before going to the next. Use of stored correct and commonly incorrect plans by scenario effectively allows an automated debriefing capability for concepts of operations and detailed plans. Use of finite state machines, both general and tied to specific scenarios, are effective for automatic debriefing of execution decisions. Providing hints, without giving away the solution, is both possible and useful.

There were a number of lessons related to the specific systems. The first was that in spite of the many standards that make interfacing separate systems easier, interfacing different hardware and software systems is often more difficult than expected, with additional data requirements not at first apparent. This is especially true if the systems being used were not originally designed for training applications.

There were several lessons relating to FBCB2. Just getting it to run successfully outside of a vehicle is difficult. In spite of their name, combat messages are not used during combat. In combat situations only 5 to 10% of the commander's time needs to be spent monitoring FBCB2's dynamic 2-D tactical display to maintain situational awareness. Using FBCB2 to create messages and plans is difficult from a practical perspective. Many of the plan symbols lack the required flexibility. The use of FBCB2 during tactical execution training is unnecessary, since most tactical simulations support a similar 2-D display.

Finally, there were several lessons relating to OTB. Most importantly, the interface is very complicated and difficult to use for training applications, since it has its own training requirements. For company commander and higher training, more intelligent unit behaviors (such as platoon leader behaviors) are needed. OTB 1.0 uses the DIS protocol, not HLA. There is a relatively small amount of available terrain for OTB. If using a 3-D viewer, the need to have additional terrain data to support the 3-D viewer means that the amount of applicable terrain is extremely small. Often there is a conflict in the way the 3-D viewer and OTB model line of sight through vegetation.

Future Work

Future work should include assessing student behavior versus Army principles/doctrine and extending the system to multiple courses. As a student takes additional courseware, a profile keeps track of weak areas and presents training missions that hit upon those areas. For example, suppose an individual tank commander, 2LT John Smith, is weak at bounding overwatch. All courses taken after the ITS assesses the student as being weak in bounding overwatch, regardless of the topic, should tend to present the student with scenarios that display skills with bounding overwatch, if possible. When taking the Armor Captain's Career Course (ACCC), the student's profile shows that bounding overwatch has finally been mastered and, therefore, relatively few of these types of scenarios are given. In ACCC, ITS notices CPT Smith does not do well at clearing minefields. Now the student gets a lot of minefield problems. For example, when the student has a mission that is primarily designed to teach breaching a berm, the ITS creates a scenario that places a minefield beyond the berm.

The ITS described here is being interfaced to the BC2010 tactical simulation and adapted for brigade and battalion commander students at the Command and General Staff College at Fort Leavenworth. This will eliminate many of the system problems described in the lessons learned section and make the combined system much more appropriate for both the CGSC and the ACCC at Fort Knox.